APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: **PEER TO PEER SOFTWARE DISTRIBUTION SYSTEM**

INVENTOR: **ROIS O. CORDOVA**

Express Mail No. <u>EL661130601US</u>

Date: <u>October 4, 2000</u>

<u>PEER TO PEER SOFTWARE DISTRIBUTION SYSTEM</u>

<u>Background</u>

This invention relates generally to the distribution of software to processor-based systems.

5   Software may be distributed from a server to a variety of client processor-based systems. Conventionally, software packages may be periodically sent to a large number of processor-based systems within a network. For example, software updates, new software, application

10   updates, and operating system updates may be distributed from a server to a large number of clients. In distributed networks, it may be necessary to implement an automated system that distributes software to a large number of clients.

15   Distributing software updates to large numbers of clients connected over a network may dramatically tax server resources. The time and resources devoted to distributing the software to a number of clients may dramatically increase the cost of the software update

20   process. This is particular so when many of the clients may be accessible through relatively low speed connections. For example, with set-top boxes that are basically processor-based systems that use television receivers as displays, relatively lower power processor-based systems

may be provided with limited bandwidth Internet connections. The software download speed may be relatively slow. As a result, the server may be taxed with slow downloads to a relatively large number of processor-based

5   systems.

Thus, there is a need for better ways to distribute software to a large number of processor-based systems.

Brief Description of the Drawings

Figure 1 is a schematic depiction of one embodiment of

10   the present invention;

Figure 2 is a flow chart for software utilized by the network management server shown in Figure 1 in accordance with one embodiment of the present invention; and

Figure 3 is a flow chart for software utilized by the

15   clients shown in Figure 1 in accordance with one embodiment of the present invention.

Detailed Description

Referring to Figure 1, a software distribution system 10 may involve the distribution of software from a network

20   management server 12 to a large number of clients 14 and 16. Each client 14 or 16 may be a processor-based system such as a desktop computer, a set-top box, a processor-based appliance or other processor-based systems. While the embodiment shown in Figure 1 includes six clients for

illustration purposes, a very large number of clients may receive software through the system 10.

In one embodiment of the present invention, a binary tree software distribution system may be utilized.

5   Initially, the network management server (NMS) 12 distributes a software package together with a list of addressees to the client 14a, over the path 12a, and the client 14b over the path 12b. Each of the clients 14 in a first tier of clients then authenticates the package by

10   providing a message back to the server 12 over a back or alternate channel 17 for example. The authentication process assures that only authorized software has been distributed to the client 14 and prevents an outside entity from attempting to disrupt the system 10.

15   Each client 14 may then distribute the received software to two clients 16 such as the client 16a and 16b in the case of client 14a or the clients 16c and 16d in the case of the client 14b. Each of the clients 16 in turn authenticates the software through the channel 17.

20   Thereafter, each of the clients 16 may in turn distribute the software to additional clients (not shown).

In this way, software may be distributed to clients that install the software and then distribute a copy of the software, together with a list of addressees, to two

25   additional clients. Each time a software receiving client 14 or 16 may delete its identifier from the list of

3

addressees, in one embodiment, so that the software is not repeatedly distributed to the same clients.

The addressee list may also be parseable. The list provided to the client 14a may be half of the original list

5   and the list provided to the client 14b may be the other half of the original list. The client 14a may then parse the list in half again and provide half of the list to the client 16a and half the list to client 16b. Alternatively, the list may be pre-parsed into a plurality of sections, as

10   received.

Each upstream, software distributing server or client identifies a receiving client to receive the software package by taking the next name on the list of addressees, in one embodiment. Thus, each client 14 or 16 may remove

15   its own identifier from the list after receiving the list from an upstream source. Each client 14 or 16 then directs the software package and the list of addressees to the next client on the list. In this way, the software package, together with the addressee list, may be progressively

20   distributed to every client in the network.

Because of the distributed or peer to peer nature of the software distribution system, the demands on the network management server 12 are reduced. The relatively longer time frame involved in downloading the software

25   package is offloaded to the clients 14 and 16. The server

12 may merely provide the needed authentication to the ultimate package recipients.

The authentication may be implemented by causing a client 14 or 16 to provide an authentication code back to the server 12, in one embodiment. If the authentication code is correct, the server 12 acknowledges that a bona fide software package was received.

In some cases, the authentication code may be successively encrypted. The code may also be modified in a known or predetermined fashion by each successive client to avoid theft by unauthorized parties. For example, a rolling code scheme may be used. The package may also be accompanied by a checksum to ensure the package was received correctly.

While a system 10 is illustrated in which each source provides a software package in a binary tree to two ensuing clients, a variety of other distribution techniques may be utilized. The number of clients that receive software from an upstream client is essentially unlimited. To the extent that the number is increased, the load on a particular upstream client software distributor is increased. Eventually, the software distribution load may be noticeable by the involved client's user. Therefore, it may be advantageous to limit the number of clients to which any one client distributes the software package.

The software for enabling the distribution of software packages may, in one embodiment of the present invention, be preloaded on each client 14 or 16. In one case, the entire system 10 is managed by a service provider that

5   controls the hardware and software provided on each client 14 or 16. Thus, the clients 14 and 16 may be provided with the necessary hardware and software (in addition to the software agents that are capable of automatically distributing the software).

10   In some embodiments of the present invention, when a software package is received, it may be immediately loaded if client resources are available to implement the loading routine. In some embodiments of the present invention, a client 14 or 16 may wait until a period of inactivity in

15   order to distribute the software to additional clients. For example, by monitoring the client's Advanced Configuration and Power Interface (ACPI) power states, such as the processor power states, the client 14 or 16 may determine an advantageous, low activity time to distribute

20   the software. In this way the software distribution is less noticeable and disruptive to the sending client's owner or user. See ACPI Specification, Revision 1.0, December 22, 1996. In another embodiment of the present invention, the software distribution may be programmed to

25   occur at night. As still another embodiment of the present

invention, the transfer time may be selectable by the client's user or owner.

In one embodiment of the present invention, the software package may include the software for the receiving client 14 or 16 as well as the software that enables the receiving client, in turn, to distribute the software to ensuing addressees on the list of addresses. In such case, not only does the client 14 or 16 receive a software package such as a software update, it also receives the software to perpetuate the software distribution system. In one embodiment of the present invention, the software distribution software may be extinguished from the client after being executed.

Turning next to Figure 2, the network management server 12 software 34 for distributing the software packages may be stored on a storage device 18. Initially, the software 34 receives a software package together with a list of client addresses as indicated at block 36. These addresses may be in a form recognized by the network, such as Transmission Control Protocol (TCP)/Internet Protocol (IP) ports and addresses as one example. The software package and the addressee list is then transported to some number of clients such as the clients 14a and 14b as illustrated in block 38. At diamond 40, the server 12 awaits an authentication request from each of the clients 14. When the authentication request is received, if the

appropriate code is provided, the software package is authenticated, for example by sending an acknowledgement, as indicated in block 42. After the required number of client addressees have received their authentication, the

5   flow may end.

In some embodiments, each client 14 or 16 may notify the server 12 that software was successfully installed on the client 14 or 16. The server 12 may then monitor the amount of time that elapses without receiving a successful

10   installation notification. If the amount of time becomes excessive, the server may check to determine if there is a problem with software installation on a particular client 14 or 16. If there is a problem, the server 12 may intervene to facilitate the installation of software on the

15   client with the problem and on other clients 14 or 16 that may ultimately encounter problems.

Referring next to Figure 3, the client software 44 may be stored on a storage 20 associated with a client 14 or 16. The software 44 is responsible for handling the

20   software package received from an upstream source and distributing the software to one or more downstream receiving clients. Initially, the software package and the addressee list is received by a given client as indicated in block 46. The client authenticates the software with

25   the network management server 12, as indicated in block 48.

The software package may then be installed on the client as indicated in block 50. Thereafter, the software package is distributed to the next name on the list of addressees as indicated in block 52. In addition, in some embodiments, the client 44 may remove its name from the list of addressees. In the case where a sending client provides the addressee list to two receiving clients, the sending client may split the list into two and provide half the list to each of the two receiving clients.

While the present invention has been described with respect to a limited number of embodiments, those skilled in the art will appreciate numerous modifications and variations therefrom. It is intended that the appended claims cover all such modifications and variations as fall within the true spirit and scope of this present invention.

What is claimed is: